# Introducción al desarrollo web (iDESWEB) - 3ª ed.

Práctica 5: JavaScript: validación de formularios

## 1. Objetivos

- Aprender el lenguaje de programación JavaScript.
- Aprender a manejar el DOM de una página web para acceder a su contenido.
- Aprender a validar un formulario con el lenguaje de programación JavaScript.

## 2. Recursos

¿Cuál es la sintaxis de JavaScript? ¿Qué palabras clave existen?

- **W3Schools**[1]: cursos de aprendizaje y guías de referencia de diversas tecnologías empleadas en la programación web. Incluye un tutorial y temas avanzados sobre JavaScript.
- **ECMA-262 ECMAScript Language Specification**[2]: estándar en el que se basa el lenguaje JavaScript.
- **Mozilla Developer Center JavaScript**[3]: información sobre JavaScript según los desarrolladores del proyecto Mozilla.
- **ECMAScript support in Opera**[4]: información sobre JavaScript según los desarrolladores del navegador Opera.
- **JavaScript Cheat Sheet**[5]: resumen de los métodos y funciones principales, la sintaxis de las expresiones regulares y el objeto XMLHttpRequest.
- **JavaScript Reference**[6]: fichas resumen de los métodos y funciones principales. Incluye una tabla con las versiones que aceptan los principales navegadores.

¿Cómo puedo comprobar que el código que escribo es correcto?

- **Firebug**[7]: extensión (*add-on*) para el navegador Mozilla Firefox, es una herramienta esencial para cualquier desarrollador web. Incluye: inspector y editor de HTML, CSS, JavaScript y DOM; regla para tomar mediciones; monitor de actividad de red y depurador de JavaScript.
- **JSLint, The JavaScript Verifier**[8]: aplicación en página web que verifica el código JavaScript, posee múltiples opciones.
- **JavaScript Lint**[9]: aplicación para descargar y ejecutar en local que emplea el motor del navegador Mozilla Firefox.
- **JavaScript Editor**[10]: potente editor de código JavaScript. Es de pago.

[1] http://www.w3schools.com/
[2] http://www.ecma-international.org/publications/standards/Ecma-262.htm
[3] http://developer.mozilla.org/en/JavaScript
[4] http://www.opera.com/docs/specs/js/ecma/
[5] http://www.addedbytes.com/cheat-sheets/javascript-cheat-sheet/
[6] http://javascript-reference.info/
[7] http://getfirebug.com/
[8] http://www.jslint.com/
[9] http://www.javascriptlint.com/
[10] http://www.c-point.com/javascript_editor.php

¿Cómo puedo esconder mi código JavaScript? No se puede, pero puedes emplear un "ofuscador" para ponérselo difícil al que quiera copiar tu código. Estas herramientas también sirven para comprimir el código JavaScript y reducir su tiempo de descarga.

- **iWeb Toolkit: Javascript Compiler**[11]: encriptador de código JavaScript.
- **Lista de ofuscadores para JavaScript**[12]: lista con enlaces a varios ofuscadores.

# 3. ¿Qué tengo que hacer?

En esta práctica tienes que emplear JavaScript para validar los formularios que has realizado en las prácticas anteriores. En concreto, tienes que programar las siguientes restricciones:

**Página principal** Contiene un formulario (nombre de usuario y contraseña) para acceder como usuario registrado. Antes de enviar el formulario, debes comprobar que el usuario ha escrito algo en ambos campos, pero evita que el usuario escriba únicamente espacios en blanco o tabuladores.

**Página con el formulario de registro como nuevo usuario** Contiene un formulario con los datos necesarios para registrarse (nombre de usuario, contraseña, repetir contraseña, dirección de email, sexo, fecha de nacimiento, ciudad y país de residencia, foto). Antes de enviar el formulario, debes realizar las siguientes comprobaciones:

- **nombre de usuario**: sólo puede contener letras del alfabeto inglés (en mayúsculas y minúsculas) y números; longitud mínima 3 caracteres y máxima 15.
- **contraseña**: sólo puede contener letras del alfabeto inglés (en mayúsculas y minúsculas), números y el subrayado; al menos debe contener una letra en mayúsculas, una letra en minúsculas y un número; longitud mínima 6 caracteres y máxima 15.
- **repetir contraseña**: su valor debe coincidir con el escrito en el campo **contraseña**.
- **dirección de email**: no puede estar vacío, hay que comprobar que cumple el patrón de una dirección de email (no permitir dominios principales de menos de 2 caracteres y más de 4 caracteres).
- **sexo**: se debe elegir un valor.
- **fecha de nacimiento**: comprobar que es una fecha válida.
- El resto de campos no indicados se pueden quedar vacíos.

# 4. ¿Cómo lo hago?

JavaScript es un lenguaje de script que comparte la misma sintaxis básica que los lenguajes de programación C, C++ y Java. El código de JavaScript se puede incluir de tres formas en una página web:

- Mediante la etiqueta `<script>`.
- En un enlace (etiqueta `<a>`) en el atributo `href`.
- En un atributo que representa un evento, como `onclick` u `onsubmit`.

Por ejemplo, la siguiente página web contiene dos líneas de código JavaScript que se ejecutan automáticamente al cargar la página web; el atributo `type` de la etiqueta `<script>` indica el lenguaje de programación empleado en el código de script:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>Prueba de JavaScript</title>
```

[11] http://www.virtualpromote.com/tools/javascript-encrypt/
[12] http://sentidoweb.com/2006/10/11/lista-de-ofuscadores-para-javascript.php

```
<script type="text/javascript">
<!--
  window.alert("Mensaje 1");
  alert("Mensaje 2");
// -->
</script>
</head>
<body>
<p>
Una página web sencilla.
</p>
</body>
</html>
```

En el ejemplo anterior, el código JavaScript está encerrado en un comentario de HTML para que no sea interpretado por aquellos navegadores que no reconocen JavaScript. Además, la llamada al método `alert()` que muestra un cuadro de diálogo con un mensaje se realiza de dos formas equivalentes: a través del objeto `window` y directamente.

Para realizar la validación de formularios lo más correcto es emplear expresiones regulares, pero eso lo haremos en la próxima práctica. **En esta práctica no tienes que utilizar expresiones regulares**. Ahora tienes que programar para cada situación el algoritmo de validación apropiado.

Existen varias formas de acceder a un formulario a través del DOM[13]:

- `document.forms[n]`, donde `n` es la posición que ocupa el formulario en la lista de formularios del documento, comenzando desde 0.
- `document.forms["nom"]`, donde `nom` es el nombre que se ha asignado al campo con el atributo `name` o `id`.
- `document.nom`, donde `nom` es el nombre que se ha asignado al formulario con el atributo `name`.

Una vez que se ha seleccionado un formulario, existen varias formas de acceder a los campos que contiene (`formu` es un objeto que representa un formulario concreto):

- `formu.elements[n]`, donde `n` es la posición que ocupa el campo en la lista de campos del formulario, comenzando desde 0.
- `formu.elements["nom"]`, donde `nom` es el nombre que se ha asignado al campo con el atributo `name` o `id`.
- `formu.nom`, donde `nom` es el nombre que se ha asignado al campo con el atributo `name` o `id`.

Una vez que se ha seleccionado un campo, se tiene que consultar la propiedad `value` para obtener el valor que almacena en un instante concreto.

Importante: este es el método tradicional de acceder al contenido de un formulario, en una próxima práctica veremos el método actual que emplea `document.getElementById()`.

En el siguiente ejemplo hay un formulario que contiene tres campos de texto para escribir el nombre de tres jugadores, el formulario no se puede enviar si no se han escrito los tres nombres:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Validación un formulario</title>
<script type="text/javascript">
function valida(f) {
  var ok = true;
  var msg = "Debes escribir algo en los campos:\n";
```

[13] Existe alguna forma más que se empleará en la próxima práctica.

```
  if(f.elements[0].value == "")
  {
    msg += "- Jugador 1\n";
    ok = false;
  }

  if(f.elements["jugador2"].value == "")
  {
    msg += "- Jugador 2\n";
    ok = false;
  }

  if(f.jugador3.value == "")
  {
    msg += "- Jugador 3\n";
    ok = false;
  }

  if(ok == false)
    alert(msg);

  return ok;
}
</script>
</head>
<body>
<form id="miForm" action="" method="get" onsubmit="return valida(this)">
<p>
Jugador 1: <input type="text" id="jugador1" />
<br />
Jugador 2: <input type="text" id="jugador2" />
<br />
Jugador 3: <input type="text" id="jugador3" />
<br />
<input type="submit" value="Enviar" />
<input type="reset" value="Borrar" />
</p>
</form>
</body>
</html>
```

En el ejemplo anterior, cuando se llama a la función `valida()` desde el evento `onsubmit` se le pasa como valor del parámetro `this`, que es un objeto que representa al formulario desde el que se invoca al método. En este ejemplo, se ha empleado el atributo `id` en vez del atributo `name` para identificar cada elemento, ya que estamos utilizando la versión Strict de XHTML 1.0, que no permite el empleo del atributo `name` en la etiqueta `<form>`.

¿Se puede usar el atributo `name` en la etiqueta `<form>` en HTML5? Consulta la recomendación oficial del W3C para averiguarlo.

La función `valida()` debe devolver `true` o `false` según la validación haya sido correcta o haya fallado. Este valor debe ser devuelto al navegador: el valor `true` significa "continúa y envía el formulario al servidor", mientras que `false` significa "cancela y no envíes el formulario". Por ello, en el código se debe escribir `onsubmit="return valida(this)"` para reenviar el valor devuelto al navegador.

## 5. Recomendaciones

Cuanta mayor separación haya entre las distintas partes que componen una página web (HTML, CSS y JavaScript) mucho mejor. Por ello, el código de JavaScript escríbelo en un fichero a parte e inclúyelo

en aquellas páginas donde lo necesites con la etiqueta `<script src=""></script>`.

En el sitio web W3Schools puedes encontrar el apartado **JavaScript Form Validation**[14] que explica algunos tipos de validación sencilla de un formulario. Además de esta página existen muchas otras más como:

- **How-To: Form validation with JavaScript**[15]: explica paso a paso como realizar una validación sencilla. Además, explica cómo validar botones de radio, casillas de verificación y listas desplegables.
- **JavaScript: Form Validation**[16]: emplea expresiones regulares para algunas de las validaciones.
- **The Art of Web: JavaScript**[17]: explica cómo realizar algunos tipos de validación especiales (contraseña, fecha y hora, tarjeta de crédito).

Para manipular las cadenas en JavaScript existe el objeto `String`. Consulta la página **JavaScript String Object Reference**[18] en W3Schools para conocer los métodos y propiedades que posee este objeto. Algunos métodos que puedes necesitar son:

- `charAt()`: devuelve el carácter situado en la posición indicada.
- `indexOf()`: devuelve la posición de la primera aparición de una cadena en otra cadena, buscando hacia delante desde la posición indicada.
- `lastIndexOf()`: devuelve la posición de la última aparición de una cadena en otra cadena, buscando hacia atrás desde la posición indicada.
- `replace()`: reemplaza unos caracteres por otros caracteres en una cadena.
- `slice()`: extrae una parte de una cadena.
- `split()`: divide una cadena en partes según los caracteres indicados.
- `toLowerCase()`: convierte una cadena a minúsculas.
- `toUpperCase()`: convierte una cadena a mayúsculas.

Para trabajar con fechas en JavaScript existe el objeto `Date`. Consulta la página **JavaScript Date Object Reference**[19] en W3Schools para conocer los métodos y propiedades que posee este objeto. Algunos métodos que puedes necesitar son:

- `getDate()`: devuelve el día del mes (de 1 a 31).
- `getMonth()`: devuelve el mes (de 0 a 11).
- `getFullYear()`: devuelve el año, como un número de cuatro dígitos.
- `getTime()`: devuelve una fecha y hora (instante de tiempo) como el número de milisegundos desde la medianoche del 1 de enero de 1970.
- `setDate()`: fija el día del mes (de 1 a 31).
- `setMonth()`: fija el mes (de 0 a 11).
- `setFullYear()`: fija el año, como un número de cuatro dígitos.
- `setTime()`: fija una fecha y hora (instante de tiempo) como el número de milisegundos desde la medianoche del 1 de enero de 1970.

[14] http://www.w3schools.com/js/js_form_validation.asp
[15] http://www.elated.com/articles/form-validation-with-javascript/
[16] http://www.webcheatsheet.com/javascript/form_validation.php
[17] http://www.the-art-of-web.com/javascript/
[18] http://www.w3schools.com/jsref/jsref_obj_string.asp
[19] http://www.w3schools.com/jsref/jsref_obj_date.asp

El objeto `Date` se puede emplear para validar una fecha introducida por el usuario. ¿Cómo? Utiliza la fecha introducida por el usuario para crear un objeto de tipo `Date` y compara la fecha introducida con los valores que devuelve el objeto.

Cuando una página contiene un error de JavaScript, los navegadores suelen informar de alguna manera al usuario. Por ejemplo, la siguiente página contiene un error, ya que el bloque de código de la función `prueba()` no está cerrado (falta una llave).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>Prueba de JavaScript</title>
<script type="text/javascript">
function prueba() {
  window.alert("Mensaje 1");
  alert("Mensaje 2");

  return false;
</script>
</head>
<body>
<p>
Un texto. Pulsa <a href="#" onclick="return prueba()">aquí</a>
para ejecutar la función.
</p>
</body>
</html>
```

En el navegador Mozilla Firefox existe en el menú Herramientas la opción Consola de errores que muestra la ventana que se puede ver en la Figura 1. En esta ventana se pueden consultar los errores de JavaScript (y de otro tipo, como CSS) que contenga una página web. En este ejemplo aparecen dos mensajes de error, uno por la llave que falta en el código y otro al haber pulsado el enlace, ya que al haber un error en el código no se ha definido la función `prueba()`.

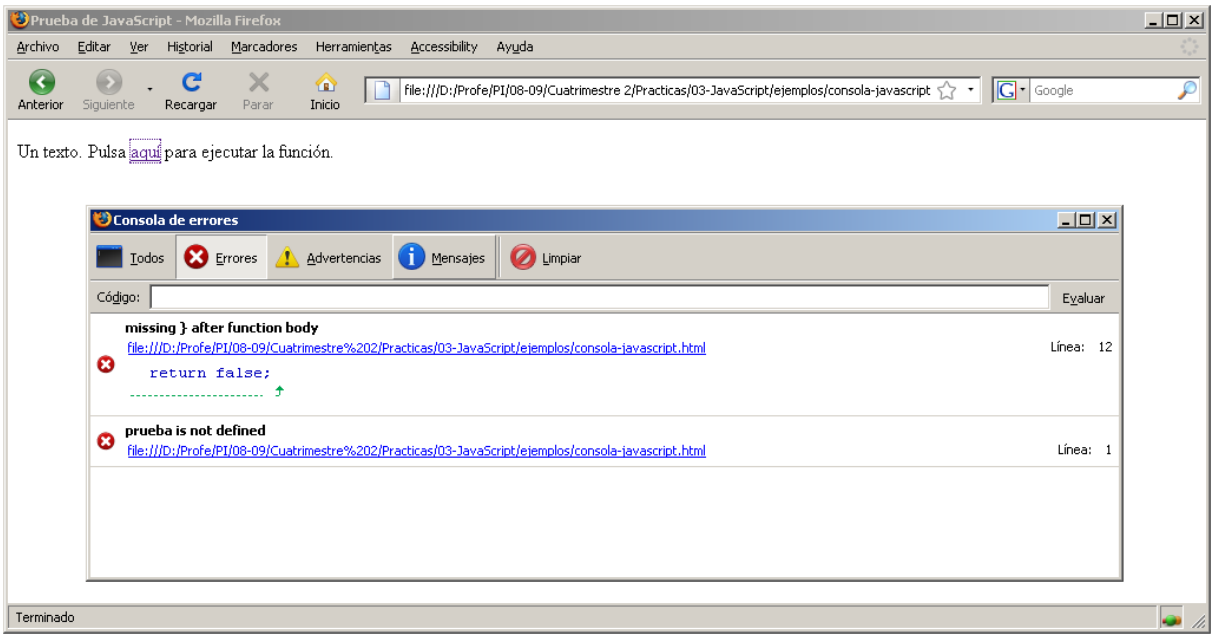


Figura 1: Consola de errores en Mozilla Firefox

En el navegador Microsoft Internet Explorer, cuando una página contiene un error de JavaScript

aparece un icono en forma de triángulo y el mensaje Error en la página en la parte izquierda de la barra de estado del navegador, tal como se puede ver en la Figura 2. Al pulsar sobre el icono dos veces, se muestra una ventana donde se especifica la localización y el tipo de error.

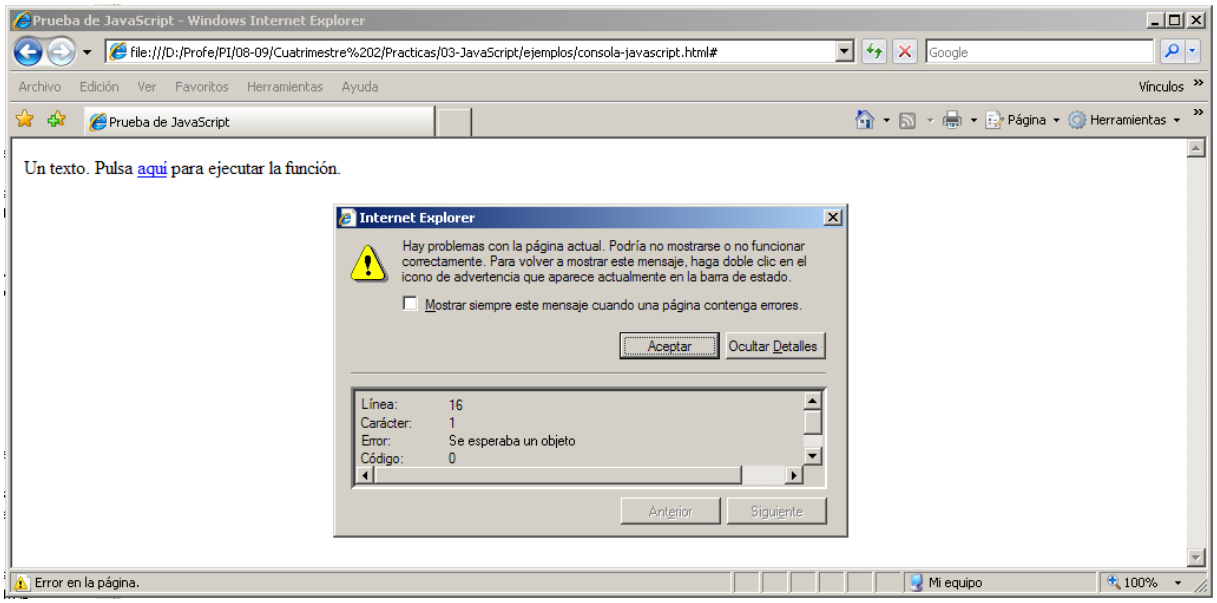


Figura 2: Ventana de errores en Microsoft Internet Explorer